A Term-Based Genetic Code for Artificial Neural Networks

Marek Musial[1] and Tobias Scheffer[2]

[1] (TU Berlin), Schottburger Str. 11 a, D-12305 Berlin, email: musia@cs.tu-berlin.de
[2] (TU Berlin), Jahnstr. 65, D-12347 Berlin, email: tobiass@cs.tu-berlin.de

1 Introduction

We developed a well-structured term-based language for structural specification of artificial neural networks. The language achieves an intuitive and compact representation even for very large networks, making it interesting as an input language for network simulators. Since functional dependencies can be expressed, well-controllable mutating operators and a useful crossover operation can be defined, thus allowing efficient optimization of network topology by genetic algorithms.

The code is based on the idea that the genotype should reflect the *logical*, schematic structure of the phenotype. A neural network might consist of *organs*, self-contained functional blocks performing a special task, maybe encapsulated in other organs on a higher structural level, and identical parts can be incorporated multiply in the network. If such abstract information is available in the genotype, mutating operators should be possible that vary the network's scheme without completely destroying the structure—or leaving it completely untouched.

In addition we wanted to define a crossover operator that preferably extracts funtionally correlated blocks of a network, without respect whether these blocks extend beyond one or more layers.

2 The Genetic Code

To achieve this, we designed a recursive, structured term-based language.

Envisaged, a term consists of a function symbol followed by a list of arguments. Depending on the symbol the arguments may again be terms, and their semantics are networks. The function symbol defines the relation of the arguments, e.g. they might be parallel or serial.

"n"-terms anchor the set. They represent n parallel units not having edges between them. The arguments of a par-term lie parallel and may themselves be structured networks. The arguments of a ser-block lie serial. The output-units of each argument are connected to the input-units of the next one. mul- and smul-terms allow "cloning" of a block n times, using shared weights in the smul-case. Thus, *feature-detectors* can be described.

The *topological filter* $\mathsf{tf}(r)$, placed between two arguments of a ser-term, determines the amount and structure of edges between these two blocks.

The step-by-step resolution of these filters allows multidimensional projections between network layers processing multidimensional input data, where the arrangement of filters defines the dimensionality assumed for the data.

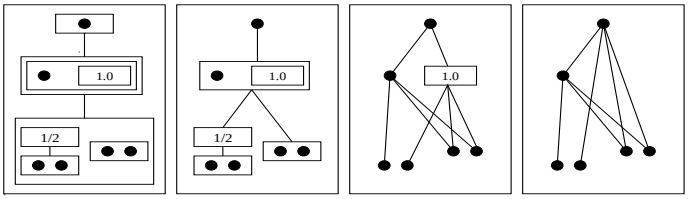

Fig.1. Resolving $\mathsf{ser}(\mathsf{par}(\mathsf{ser}(2, \mathsf{tf}(1/2)), 2), \mathsf{par}(\mathsf{par}(1, \mathsf{tf}(1.0))), 1)$

The terms can be translated into sentences of a diagram language. By applying diagram transformation rules, blocks are resolved and edges inferred. The result of this process is a neural network. Figure 1 shows an example resolution.

3 Mutation & Crossover

For usage as a genetic code, mutating operators and crossing-over have to be defined on the sentences of the language.

The mutating operators can be defined closely to the phenotype level. Structure-preserving mutating operators of different levels of abstraction are possible. Crossing-over now means term-replacement.

4 Results

Even very large neural networks can be represented in an intuitive and compact way. The ZIP-code-reader by Le Cun et al. [1], consisting of about 1000 Units, can be encoded in less than three lines. Compared to Koza and Rice [2], we achieve a higher level of abstraction. Our genetic algorithm found network topologies for multiple problems, e.g. the two-spirals problem [4] and real-valued functions.

References

1. Y. Le Cun, B. Boser, J. S. Denker, D. Hendersen, R.E Howard, W. Hubbard, L. D. Jackel: "Backpropagation Applied to Handwritten Zip Code Recognition"; *Neural Computation* 1, pp 541, 1989
2. J. R. Koza, J. P. Rice: "Genetic Generation of both the Weights and Architecture for a Neural Network"; *IJCNN92*, pp 397, 1991
3. Frédéric Gruau: "Cellular Encoding as a Graph Grammar"; *ISM*, 7/93
4. Scott E. Fahlmann, Christian Lebiere: "The Cascade-Correlation Learning Architecture", *Neural information processing Systems 2*, 1990

This article was processed using the LaTeX macro package with LLNCS style